

Localization in Non-localizable Sensor and Ad-hoc Networks: A Localizability-aided Approach

Tao Chen*, Zheng Yang^{†‡}, Yunhao Liu^{†‡}, Deke Guo^{*§} and Xueshan Luo*

*College of Information System and Management, National University of Defense Technology, China

[†]TNLIST, School of Software, Tsinghua University, China

[‡]Department of CSE, Hong Kong University of Science and Technology, Hong Kong

[§]School of Computer Science and Technology, Huazhong University of Science and Technology, China

Abstract—Localization is an enabling technique for many sensor and ad-hoc network applications. Real-world deployments demonstrate that, in practice, a network is not always entirely localizable, leaving a certain number of theoretically non-localizable nodes. Previous studies mainly focus on how to tune network settings to make a network localizable; however, they are considered to be coarse-grained, since they equally deal with localizable and non-localizable nodes. Ignoring localizability induces unnecessary adjustments and accompanying costs. In this study, we propose a fine-grained approach, Localizability-aided Localization (LAL), which basically consists of three phases: node localizability testing, component tree construction, and network adjustment. LAL triggers a single round adjustment, after which some popular localization methods can be successfully carried out. Being aware of node localizability, all adjustments made by LAL are purposefully selected. Simulation results show that LAL effectively guides the adjustment.

I. INTRODUCTION

As the proliferation of wireless and mobile devices continues, a wide range of context-aware applications are deployed, including smart space, modern logistics, etc. In these applications, location information is the basis of other services, such as geographic routing, boundary detection, and network coverage control. A number of approaches [1]–[3] have been proposed for in-network localization, in which some special nodes (called beacons or anchors) know their global locations and the rest determine their Euclidean coordinates by measuring the Euclidean distances to their neighbors. The localization problem is to figure out the locations of other nodes based on inter-node distance measurements and global locations of beacons.

Due to hardware or deployment constraints, a network can be partially localizable, that is to say, some nodes have unique locations while others do not. We find supporting evidences from two working sensor network systems OceanSense [4] and GreenOrbs [5]. Networks are not entirely localizable in their initial deployments. Based on graph rigidity theory, network localizability [6] and node localizability [7] are studied which solve the location-uniqueness of a network or a single node, respectively. So far one can distinguish localizable and non-localizable nodes in theory. To locate non-localizable nodes, a practical and convenient way is to increase the distance ranging capability of sensor nodes. Enhanced nodes can measure the distances to a larger number of their nearby nodes, thus enhancing localizability. In popular ranging techniques, such

as Time of Arrival (ToA) and Received Signal Strength (RSS), the capability enhancement can be achieved by augmenting the transmitter power output. Taking a typical sensor node TelosB with CC2420 chip as example, the RF output power is programmable and divided into 31 levels, and the maximum ranging distance varies from tens of centimeters to over one hundred meters. Anderson *et al.* [8] propose an approach to construct localizable networks based on power augmentation. In their method, nodes are required to set their power doubly or triply of the original values. We consider the method is coarse-grained, since it equally treats localizable and non-localizable nodes. Ignoring localizability induces a mass of meaningless adjustments and accompanying costs.

In this study, we propose Localizability-Aided Localization (LAL), which basically consists of three phases: node localizability testing, component tree construction, and network adjustment. We adopt the sufficient condition of node localizability [7] in the first phase to identify localizable and non-localizable nodes. Then a fine-grained configuration is computed systematically according to network topology and localizability information in the second and third phases. LAL triggers a single round adjustment, which does not require multiple round of configuration dissemination and data collection. After that, some popular localization methods can be successfully carried out. Theoretical analysis guarantees a tuned network is localizable and thus well-prepared for localization. The main contributions of this study are as follows: First, being aware of node localizability, adjustments made by LAL are purposefully selected, avoiding meaningless ranging and communication costs. Second, LAL is light weight, working properly with existing localization methods (e.g., iterative trilateration, Robust Quad [3], Sweeps [2], etc.) without incompatibility. Finally, extensive simulations are conducted to validate the effectiveness of our design.

The rest of the paper is organized as follows. We provide the theoretic foundations for LAL in Section II. Section III presents the design and implementation of LAL. Evaluation is discussed in Section IV. We conclude the work in Section V.

II. THEORETIC FOUNDATION

A. Preliminary

The ground truth of a network can be modeled by a distance graph $G = (V, E)$, where V denotes the set of vertices

(wireless devices, e.g. sensor node, RFID) and E denotes the set of edges. For $i, j \in V$, $(i, j) \in E$ if the distance between i and j can be measured or both of them are in known locations (e.g. beacon nodes). We assume G is connected and has at least four vertices, and focus on 2D space in the following analysis.

Localization and localizability are usually studied in the framework of graph theory. Localization is to find a map p (a.k.a., *realization*) from vertices in G to points in a Euclidean space. If p respects the distance constraints between any pair of vertices as depicted in E , p is a feasible realization of graph G . Two realizations are equivalent if they are identical under translations, rotations and reflections. A distance graph G has at least one feasible realization which represents the ground truth of the corresponding network.

A realization is *generic* if the vertex coordinates are algebraically independent. Since the set of generic realizations is dense in the realization space, almost all realizations are generic and we omit this word hereafter. A graph is called *rigid* if one cannot continuously deform its realizations while preserving distance constraints, and *redundantly rigid* if it remains rigid upon removal of any single edge [9]. A graph is *globally rigid* if it is uniquely realizable [10].

Theorem 1 ([11]). *A graph with $n \geq 4$ vertices is globally rigid in 2 dimensions if and only if it is 3-connected and redundantly rigid.*

A graph $G = (V, E)$ is called k -connected (for $k \in \mathbb{N}$) if $|V| > k$ and $G - X$ is connected for every set $X \subseteq V$ with $|X| < k$. In other words, no two vertices in G are separated by fewer than k other vertices.

Eren *et al.* [6] further prove that a network is uniquely localizable if and only if its distance graph is globally rigid and it contains at least three beacons.

Theorem 2 ([7]). *In a distance graph $G = (V, E)$ with a set $B \subset V$ of $k \geq 3$ vertices at known locations, a vertex is localizable if it is included in the redundantly rigid component inside which there are three vertex-disjoint paths to three distinct vertices in B .*

A redundantly rigid component is a maximal redundantly rigid subgraph in G . Theorem 2 is a sufficient condition of node localizability, denoted by RR3P for short, where RR and 3P stand for redundant rigidity and three vertex-disjoint paths, respectively.

Given a non-localizable graph, it is important to make it localizable through incremental construction. Define G^2 as $(V, E \cup E^2)$ where $(i, j) \in E^2$ if $i \neq j$ and $\exists k \in V$ such that $(i, k) \in E$ and $(j, k) \in E$. Similarly, define G^3 as $(V, E \cup E^2 \cup E^3)$ where $(i, j) \in E^3$ if $i \neq j$ and $\exists k \in V$ such that $(i, k) \in E$ and $(j, k) \in E^2$. Based on Theorem 1, Anderson *et al.* present the following result.

Theorem 3 ([8]). *Let $G = (V, E)$ be a 2-edge-connected graph in 2-dimension space. Then G^2 is globally rigid.*

G is called l -edge-connected, if $|V| > 1$ and $G - F$ is connected for every set $F \subseteq E$ of fewer than l edges. A k -

connected graph is also k -edge-connected.

G^2 is obtained from G by adding edges between two-hop vertex pairs. Note that in some ranging techniques (e.g. RSS and ToA), RF signal is used for distance measurement. Thus the communication range approximately represents the maximum ranging distance. In a wireless network, doubling the communication range will produce a new graph which has G^2 as a subgraph. Similarly, G^3 can be obtained by tripling the communication range. Anderson *et al.* [8] also prove that, if $G = (V, E)$ is connected, G^3 is globally rigid (specifically, G^3 can be localized by iterative trilateration).

B. Component Tree

To support fine-grained network adjustment, we decompose a graph into components according to connectivity. A 2-connected component in a graph G is a maximal subgraph of G without *articulation* vertex whose removal disconnects G . We obtain the following theorem about 2-connected component and localizable vertices.

Theorem 4. *Given a connected distance graph G with more than 3 vertices, all beacons and localizable vertices recognized by RR3P in G are in the same 2-connected component.*

Due to space limit, we omit the detail proof hereafter. We call the 2-connected component in Theorem 4 *anchor component* (denoted by G_A) for convenience.

Theorem 5. *Suppose $G = (V, E)$ is a 2-connected graph with a set B of $k \geq 3$ beacons and $B \subset V$. Let V_N denote the set of non-localizable vertices, and E_N denote the set of edges (i, j) , $i \in V_N$ and $(i, j) \in E^2$. Then, $G = (V, E \cup E_N)$ is localizable.*

A direct result of Theorem 5 is that, if we add E_N to the anchor component, G_A becomes localizable. G_A can be seen as localizable in the following analysis.

We decompose a graph into 2-connected components. Except for the anchor component, all of them are non-localizable. We classify the components into two categories, as illustrated in Fig. 1. Category I has two inter-connected vertices, and is trivially 2-connected. Category II has at least three vertices. The anchor component is a category II component.

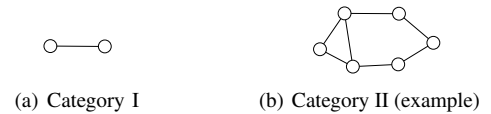


Fig. 1. Two categories of 2-connected components.

Two components sharing a common vertex are adjacent. They cannot have more than one common vertex, or they should be merged into one. We construct component tree $T = (V_T, E_T)$ of $G = (V, E)$ as follows. Any vertex $i \in V_T$ corresponds to a 2-connected component of G and $(i, j) \in E_T$ if the correspondents of i and j are adjacent in G . For convenience, let G_A be the root of T . We decompose G into

several 2-connected components and connect neighboring ones in T . Following such construction, T contains no cycle.

Location information, as well as localizability, diffuses from the root of T to leaf vertices along tree edges. Another issue needs to be addressed here is the path from the root to any tree vertex in T . The category permutation of the components along a path determines how to make adjustments in our solutions. For simplicity, we use a string such as “II+...+I” to describe the categories of components on a path, from the root to a destination component. We call such a string “path string” henceforth.

C. Component Based Adjustment

Before discussing the detail of component based adjustment, we define the operation of vertex augmentation.

Definition 1. (Vertex Augmentation) In a graph $G = (V, E)$, the vertex augmentation of $v \in V$ (denoted by v^k , $k \in \mathbb{N}$) is to connect v and its i -hop neighbors in G ($i \leq k$).

In a distance graph $G = (V, E)$ of a network, for any $v \in V$, v^n can be achieved by increasing the ranging capability of v (more accurately, v 's correspondents) by n times.

Since there are two categories of 2-connected component according to the classification in this work, the path string of a component can be decomposed into a series of overlapping segments of two consecutive categories. There are four cases: “I+I”, “I+II”, “II+I”, and “II+II”. If we provide adjustment solutions for all four cases to make the corresponding components localizable, the entire network becomes localizable by applying these solutions from the root to leaves in the component tree. When discussing the solutions of the four cases, we assume the former component has already been manipulated and thus localizable. The assumption is reasonable because G_A is the root and made localizable according to Theorem 5.

• Case “II+I”

Suppose two corresponding components are G_1 and G_2 , as shown in Fig. 2. The vertex v_1 is localizable and has two localizable neighbors v_2 and v_3 in G_1 . Suppose the neighbor of v_1 in G_2 is v_4 . The operation v_4^2 adds two edges (v_4, v_2) and (v_4, v_3) in E^2 , which makes v_4 localizable by connecting it to three localizable vertices.

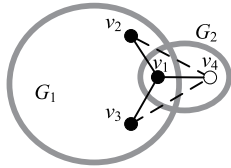


Fig. 2. Case “II+I”: localize the category I component by adding two edges. Solid lines are edges in E and dashed lines are edges in E^2 .

• Case “II+II”

Fig. 3 shows two category II components G_1 and G_2 . The vertex v_1 is localizable, and has two localizable neighbors v_2 and v_3 in G_1 , and two non-localizable neighbors v_4 and v_5 in G_2 . G_2^2 is globally rigid according to Theorem 3. Similar

to the case “II+I”, v_4 and v_5 are made localizable by v_4^2 and v_5^2 . Then G_2^2 is localizable since it is globally rigid and has three vertices whose location can be uniquely determined by calculations.

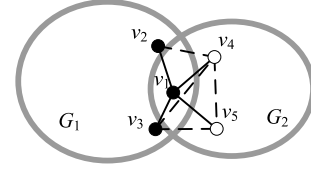


Fig. 3. Case “II+II”: G_2^2 is localizable through v_4^2 and v_5^2 . Solid lines are edges in E , and dashed lines are edges in E^2 .

• Case “I+I”

Suppose the two category I components are G_2 and G_3 . v_2 is the articulation vertex of G_2 and G_3 , and two vertices v_1 and v_3 reside in two components, respectively, as shown in Fig. 4. Because all paths start with a category II component G_A , G_2 must have a parent component G_1 . No matter the category of G_1 , v_1 has at least one localizable neighbor v_4 . v_2^3 only adds a single edge (v_1, v_3) , and v_3 still does not fulfill the sufficient condition in Theorem 2. To make v_3 localizable, at least one edge such as (v_3, v_4) should be added. That is to say, we should do v_3^3 in the original topology to make G_3 localizable.

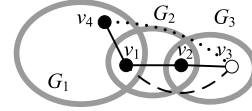


Fig. 4. Case “I+I”: v_3^3 is needed to make G_3 localizable. Solid lines are edges in E . The dashed line is an edge in E^2 , and the dotted line is an edge in E^3 .

• Case “I+II”

Suppose the two components are G_2 and G_3 as shown in Fig. 5. Similar to the case “I+I”, v_3^3 is needed to make v_3 localizable. Then, v_4^2 makes v_4 localizable. Since G_2^3 is globally rigid according to Theorem 3 and the locations of three vertices v_2 , v_3 , and v_4 can be uniquely determined, all vertices in G_3 are localizable.

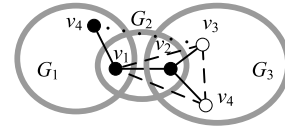


Fig. 5. Case “I+II”: v_3^3 (or v_4^3) is needed to make G_3^2 localizable.

III. LAL DESIGN AND IMPLEMENTATION

The previous section established the theoretical foundation of LAL. In this section, we discuss LAL design and implementation. Basically, LAL consists of three major steps, as depicted in Fig. 9.

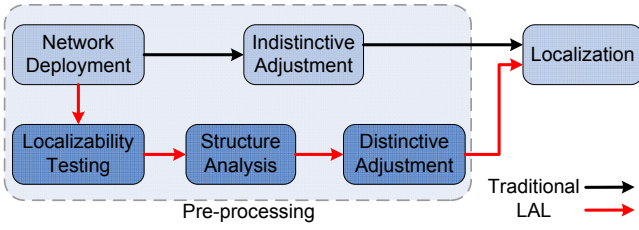


Fig. 6. Workflows of traditional approaches and LAL.

- Step 1: Localizability testing. When a network is deployed in an application field, due to some systematic or environment factors unpredictable in the design phase, it may be not ready for localization. Hence, node localizability is conducted foremost in LAL, which identifies localizable and non-localizable nodes in a network for further adjustment.
- Step 2: Structure analysis. To support fine-grained manipulation, we decompose a distance graph into 2-connected components. These components are organized in a tree structure and the one containing beacons is the root. Adjustments are conducted along tree edges from the root to leaves.
- Step 3: Distinctive adjustment. LAL treats nodes differently according to their localizability and places in the component tree. Through vertex augmentation, LAL converts all non-localizable in one round. The networks tuned by LAL are localizable and can be localized by existing localization approaches.

Step 1 can be done by applying Theorem 2. Given a specific node, its localizability relies on the property of disjoint paths and redundant rigidity, which can be tested in polynomial time by network flow algorithms and the pebble game algorithm [12], respectively. In step 2, a graph is decomposed into 2-connected components using depth-first search. In step 3, node adjustments described in Section II are conducted along the paths of the component tree starting at the root. For step 3, we propose an algorithm *LAL_Basic* as described below.

In *LAL_Basic*, edges are first added by vertex augmentation of all non-localizable vertices in G_A , and G_A is then localizable according to Theorem 5. For a component other than G_A , the categories of the component and its parent component are used to guide adjustments as described in Section II. The above process is first applied to the children components of G_A , and recursively repeats until all components get manipulated. After applying the above algorithm, the entire network is localizable. Popular localization algorithms can then be used seamlessly to localize all nodes in the network without compatibility.

Proposition 1. *Given the locations of localizable nodes, for any non-localizable node v in a connected graph $G = (V, E)$, v can be localized by Sweeps [2] after the execution of *LAL_Basic*.*

To reduce the number of redundant edges added in category II component, we analyze the graph properties of these components and find the following observations in the original

Algorithm 1 Add_Heuristic

Require: A 2-connected component G_i , and localizability vector of vertices.

```

1: if  $G_i$  is  $G_A$  then
2:   for each non-localizable vertex  $v_j \in G_i$  do
3:     if  $v_j$  has localizable neighbor then
4:       Add  $v_j$  into a set  $V_D$ .
5:     end if
6:   end for
7:   for each vertex  $v_j \in V_D$  do
8:     Add edges based on Observation 1.
9:     Mark  $v_j$  localizable.
10:    Apply Observation 3.
11:   end for
12: end if
13:  $flag \leftarrow 1$ 
14: while not all vertices in  $G_i$  are localizable and  $flag == 1$  do
15:    $flag \leftarrow 0$ 
16:   for non-localizable vertex  $v_j \in G_i$  do
17:      $num \leftarrow$  amount of localizable 2-hop neighbors
18:     if  $num \geq 3$  then
19:       Add edges based on Observation 2.
20:       Mark  $v_j$  as localizable.
21:        $flag \leftarrow 1$ 
22:       Apply Observation 3.
23:     end if
24:   end for
25: end while
26: for each vertex  $v_k$  marked non-localizable in  $G_i$  do
27:   Vertex augmentation  $v_k^2$ 
28: end for

```

communication graph.

Observation 1: In G_A , some non-localizable vertices have one localizable neighbor. The localizable vertex has at least three vertex-disjoint paths to three beacons. Adding two edges which connect two neighbor vertices (of the localizable one) on different vertex-disjoint paths to the non-localizable vertex is enough to make the vertex localizable. Moreover, only one edge is needed if a non-localizable vertex has two localizable neighbors in G_A .

Observation 2: If a non-localizable vertex has three or more localizable vertices within two-hop distance, connecting it to three localizable vertices makes it localizable.

Observation 3: Some globally rigid components are not localizable in the original network topology due to the lack of beacons. If three nodes are adjusted to be localizable in a globally rigid component, the component is immediately localizable without extra manipulation.

We revise *LAL_Basic* and propose a heuristic algorithm *LAL_Heuristic*. The difference between the two algorithms lies in the way they add edges in category II components. *LAL_Heuristic* uses a function named *Add_Heuristic* to add edges instead of vertex augmentation. The pseudo-code of

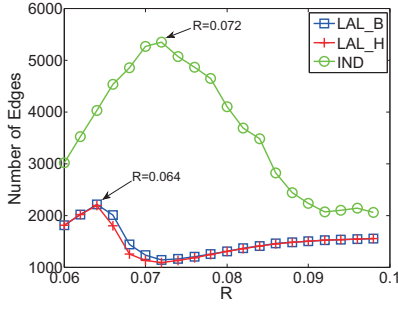


Fig. 7. Number of edges in the graph after adding edges.

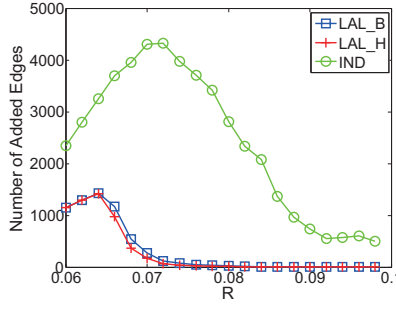


Fig. 8. Number of added edges.

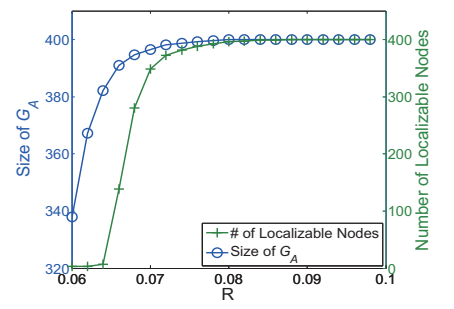


Fig. 9. Number of localizable nodes and nodes in anchor component.

the function is shown in Algorithm 1. The correctness of Algorithm 1 is guaranteed by Theorem 1 and Theorem 5.

IV. EVALUATION

We compare LAL with the indistinctive approach proposed by Anderson *et al.* [8], denoted by IND. LAL_B and LAL_H denote *LAL_Basic* and *LAL_Heuristic*, respectively. We randomly generate networks of 400 nodes which are uniformly distributed in a unit square $[0, 1]^2$. The disk model with a radius is adopted for communication and distance measurement. For each setting, we integrate results from 100 network instances, with three beacons in each instance. The original communication range is denoted by R . For each instance, we change the value of R from 0.06 to 0.98 stage-by-stage with a step length of 0.002.

As shown in Fig. 7 and Fig. 8, our approach needs to add much less edges to the original distance graph than IND does. The largest difference is found when $R = 0.072$, where IND adds about 5000 more edges than LAL_B and LAL_H. The variation of all the curves in Fig. 7 and Fig. 8 can be explained with Fig. 9. When the value of R is small, the network is not 2-connected in most instances. IND needs to triple the communication range of all nodes in these instances. As R increases, most network instances are still not 2-connected, but the average degree is increased, so the number of edges keeps increasing. When R increases to a larger value such as 0.074, the average size of G_A are over 395, which implies most instances are 2-connected. Nodes in 2-connected instances only need to double the range, and at the same time more instances become localizable, so the number of edges of IND decreases as the value of R increases. When $R > 0.085$, most instances has no non-localizable nodes and no edges need to be added. The curves of LAL_B and LAL_H follow the same trend. They reach the peak value when $R = 0.064$, because R and the number of edges are keep increasing while the number of localizable nodes in G_A keeps almost unchanged with a very small value as shown in Fig. 9. We can also find that LAL_H needs fewer extra edges than LAL_B does in most cases.

V. CONCLUSION

We analyze the limitations of existing studies on localization in non-localizable networks, and propose a localizability-aided

approach named LAL. LAL make adjustments according to node localizability results, other than indistinctively consider the network as a whole. Our designs not only excel previous work theoretically, but also have some good characteristics for practical application.

ACKNOWLEDGEMENT

This work is supported in part by the NSF China under Grants No. 60970118, 60803152, 60903206, 61070216, the Preliminary Research Foundation of China under Grant No. 9140A06050610KG0117, the China Postdoctoral Science Foundation under Grant No. 20100480898, and the Preliminary Research Foundation of National University of Defense Technology.

REFERENCES

- [1] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system", in *Proceedings of ACM MobiCom*, 2000.
- [2] D. K. Goldenberg, P. Bihler, M. Cao, J. Fang, B. D. O. Anderson, A. S. Morse, and Y. R. Yang, "Localization in sparse networks using sweeps", in *Proceedings of ACM MobiCom*, 2006.
- [3] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements", in *Proceedings of ACM SenSys*, 2004, pp. 50–61.
- [4] "Oceansense project", <http://www.cse.ust.hk/liu/Ocean/index.html>.
- [5] L. Mo, Y. He, Y. Liu, J. Zhao, S.-J. Tang, X.-Y. Li, and G. Dai, "Canopy closure estimates with greenorbs: sustainable sensing in the forest", in *Proceedings of ACM SenSys*, 2009, pp. 99–112.
- [6] T. Eren, D. K. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. O. Anderson, and P. N. Belhumeur, "Rigidity, computation, and randomization in network localization", in *Proceedings of INFOCOM*, 2004.
- [7] Z. Yang and Y. Liu, "Understanding node localizability of wireless ad-hoc networks", in *Proceedings of INFOCOM*, 2010.
- [8] B. D. Anderson, P. N. Belhumeur, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, and Y. R. Yang, "Graphical properties of easily localizable sensor networks", *Wirel. Netw.*, vol. 15, no. 2, pp. 177–191, 2009.
- [9] G. Laman, "On graphs and rigidity of plane skeletal structures", *Journal of Engineering Mathematics*, vol. 4, pp. 231–340, 1970.
- [10] B. Hendrickson, "Conditions for unique graph realizations", *SIAM Journal of Computing*, vol. 21, no. 1, pp. 65–84, 1992.
- [11] B. Jackson and T. Jordán, "Connected rigidity matroids and unique realizations of graphs", *J. Comb. Theory Ser. B*, vol. 94, no. 1, pp. 1–29, 2005.
- [12] D. J. Jacobs and B. Hendrickson, "An algorithm for two-dimensional rigidity percolation: the pebble game", *Journal of Computational Physics*, vol. 137, pp. 346–365, 1997.